

# Development and Prototyping of Android Vehicle Maintenance Applications using Figma and Kotlin

Ajub Ajulian<sup>1</sup>, Denis<sup>2</sup>, Achmad Hidayatno<sup>3</sup>, Sukiswo<sup>4</sup>

<sup>1,2,3,4</sup> Lecturer, Department of Electrical Engineering, Diponegoro University, Central Java, Indonesia

## ABSTRACT

Android application development has been increasing for several years because there is a need to build a platform that could become a place that meets the seller and consumer. Many sectors could be helped by developing a special-purpose application to support the company's service. One of the sectors is transportation and vehicle companies. There is a maintenance process that is needed in every vehicle. Because people have become busy, they sometimes forget to maintain their cars. A problem could be solved by developing an Android with an alarm feature to remind the user to do vehicle maintenance. The application could save information about the service's history and if there is a defect in the spare part of the vehicle. Suppose the record could be maintained in an Android application using a connected database. Hopefully, machine learning could predict based on the user's behaviour to do maintenance. This research does not discuss machine learning but the development methodology that has been done in Android applications. The application could run well and give the user more awareness because the alarm works fine.

**Keywords:** Android application, vehicle maintenance, development

## 1. INTRODUCTION

Applications must be more effective in performance, appearance, and use in the modern digital world. Due to the ease they provide and the fact that there are 2.7 billion active users of Android applications globally, they are one of the most widely used forms of media being generated right now. Many mobile applications are being developed that undoubtedly assist individuals in their daily routines, starting with applications that specifically benefit people, such as auto maintenance, online shopping, and reminders.

It is also evident that Indonesia is becoming denser in terms of motorized vehicles, particularly cars. However, this trend is not always directly correlated with driver awareness of the state of their vehicles. The safety of its users may be indirectly affected by those currently less concerned with the upkeep of their automobiles. A damaged accumulator, insufficient engine oil, a stuck clutch, a problem or breakdown in the middle of the road, and even more fatal accidents like accidents are examples of events that could happen.

Based on data from the division of traffic division of the police department published by the Ministry of Transportation, the number of traffic accidents in Indonesia will reach 103,645 cases in 2021. This number is higher than the 2020 data of 100,028 cases. In most cases, the cause of an accident is the vehicle itself, such as the engine's condition, brakes, lights, tires, and load. The vehicle's poor condition is caused by vehicle owners who forget or don't care about the maintenance of their vehicle used daily.

This research's primary aim was to build a vehicle maintenance application to track the vehicle's health that the user has added. Users of this application can add their vehicles and submit data in the form of the most recent maintenance date. Users of the program can recommend when maintenance should be performed using the data provided and provide other information by presenting garages near locations that can be visited to perform vehicle maintenance. From a commercial standpoint, this application will implement a system of partnerships through which it will sponsor workshops partnered with the application to earn income.

## 2. THEORETICAL FOUNDATION

### 2.1 Unified Modeling Language

Object-oriented systems are designed using the visual modelling technique known as UML (Unified Modeling Language). The Object Management Group initially developed UML, which had its first version of 1.0 in January 1997. UML is often a standard language for creating software blueprints and can be used to visualize, design, and document systems [1].

UML could provide users with a visual or drawing modelling language of a wide variety of programming and general process engineering. UML brings together the best available information in modelling. UML provides an overview of the model or as an expressive visual modelling language in system development. UML describes not only the software system model but can also model object-oriented systems. UML makes users easier to read an outline. UML is useful as a blueprint that will explain more detailed information in the design by coding a program.

The following list outlines the requirement for UML and its function:

1. It can offer users visual modelling or drawing language for various programming and general process engineering.
2. It combines the finest data that is currently available for modelling.
3. Describe the model in general or serve as a powerful visual modelling language throughout system development.
4. It could model object-oriented systems in addition to describing the model of the software system.
5. Make a system more accessible for users to understand.
6. This model will explain more specific information about the design in the form of program coding, making it useful as a blueprint.

### 2.2 Android Operating System

The popular operating system Android has been incorporated into many of today's technology. Android was an operating system created especially for smartphones and tablets. Numerous technology businesses have used Android as the operating system for their creations. According to Dicoding, around 90% of mobile devices in Indonesia run on the Android operating system. Because of this, several businesses, including Youtube, Facebook, and Twitter, have created Android versions to be more accessible to their users. Android applications are also necessary for small businesses to tackle challenges now [2].

### 2.3 Android Application Development

Various development procedures are used in Android application development to create applications that can run on devices running the Android operating system. According to sources on the Amazon website, Mobile Device Application Development (Programming) is the process of developing software applications that operate on mobile devices [3]. Typical mobile applications use a network connection to interact with external computing resources. The method of producing an Android application consists of the following eight steps, which a developer must complete to create an appealing application:

#### a. Analysis of User Needs

At the needs analysis stage, the definition of the problem is carried out. At this stage, it is estimated that all possible problems might arise in a system. Requirements analysis also aims to discover all potential problems in the system.

#### b. Planning

This stage is carried out so that the process of making software in the future will be more efficient. The most critical step in development is building an application because whatever is planned will be the basis for developing the application.

#### c. Application Design

A UI/UX design will be made for the application at this stage. The data obtained from the need analysis is used as a reference in making the design.

#### d. Program Development

A variety of procedures and steps are involved in application development. Setting up actual storage, databases, APIs, and server solutions is crucial for the back end of applications. A developer account for the app store you intend to utilize to distribute your app is also vital.

#### e. Documentation

A crucial step in the software development process that shouldn't be overlooked is documentation. The documentation's purpose or advantage serves as a roadmap for the software development process. Additionally, documentation simplifies explaining the software to other customers or teams.

#### f. Testing Applications

Once the app has a complete concept with attractive graphics and properly placed content, testing can begin. Conducting application testing helps find technical weaknesses or bugs and fix them immediately. Developers can check the initial planning and design documents and crosscheck everything with the app model to see if the requirements have been met successfully.

#### g. Deployments

The stage of making software does not stop at testing, even though the application can run well. After that, there is still the deployment process. In the deployment process, it is necessary to deploy the application on the server and determine whether it meets the specified security standards for data leakage.

#### h. Updating and Maintenance

Updating and maintaining software is the final step in the process. Launched applications cannot be left unattended. The created application has to have upkeep or maintenance performed by the developer.

### 2.3 Figma

Designing interfaces that can be shared in real-time with other teams can be done using the collaborative online application Figma. One benefit of this program is that just the free features can fulfil the needs of beginning designers. Additionally, Figma can be used straight in a web browser without installing programs. The feature set of Figma is centred on designing user interfaces and user experiences. The Figma application also makes use of various prototyping and vector graphics editors [4].

Figma can be used as a supplementary tool to assist in application design. A UI/UX designer can more readily create plans for an application, from wireframes to prototypes, using Figma. Due to the real-time editing features of Figma, a UI/UX designer can collaborate with the team without difficulty. Additionally, several plugins and templates exist that can expedite the design process.

### 2.4 Kotlin

JetBrains created the programming language Kotlin as part of an open-source undertaking in 2010. A statically typed programming language with support for the JVM, Android, JavaScript, and Native, Kotlin is also available. Because it uses the LLVM compiler, this programming language may be converted into JavaScript code. Kotlin can construct various applications, including server or backend, web, and Android mobile. Kotlin, still in development, enables programmers to create programs for additional platforms such as embedded systems, PCs, macOS, and iOS [5].

### 2.5 Android Studio

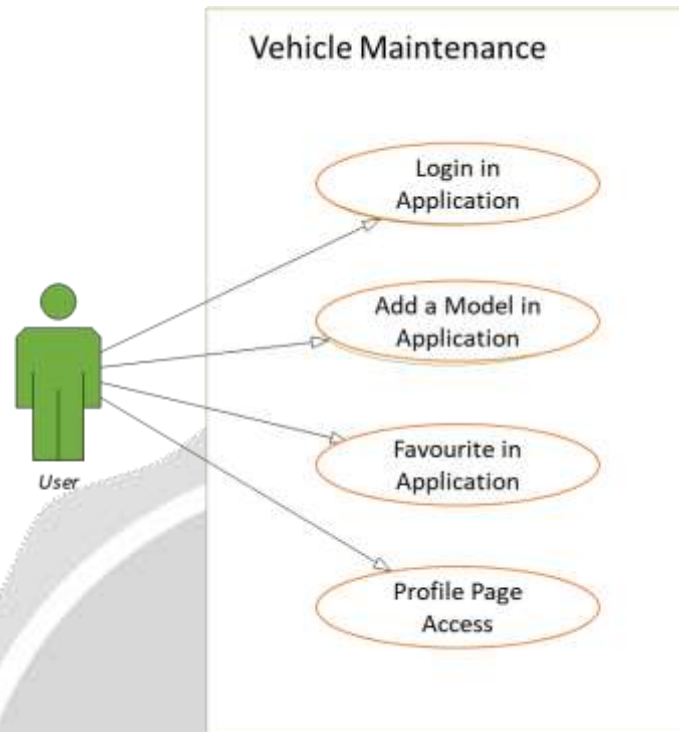
Based on IntelliJ IDEA, Android Studio is the official Integrated Development Environment (IDE) for creating Android applications. It is free and open-source software. Android Studio took the role of Eclipse as the official IDE for developing Android applications after its release, which was announced on May 16, 2013, at the Google I/O Conference. Android Studio combines a build system, emulator, code templates, and Gradle-based connectivity to Github to assist application development for the Android OS [6].

### 2.6 Github

Git is the parent version of Github, a Software Version Control System. A version Control System, commonly abbreviated as VCS, is the best tool for collaboration between developers when building applications. Big companies have used it to manage their products or applications [7].

## 3. METHODOLOGY

For describing interactions between actors and systems in this context, use case diagrams are helpful. As seen in Figure 1, a user rather than a vehicle maintenance application is the entity at stake in this use case. The vehicle maintenance application's user can access several features, including logging in, creating an account, registering and filling out car details, applying for a preferred automobile, and accessing the profile page to modify the language or log out.



**Fig -1** Vehicle Maintenance Use Case

Activity flow diagrams show how each activity flow in the system under design starts, how decisions might be made along the way, and how it ends. An activity diagram from the vehicle maintenance application is shown below. Users can log in by entering their email addresses and password. If they have already registered; otherwise, they can use the "Register Account" page if they do not already have an account or the "Forgot Password" page if they have forgotten their account password. The user can log in to the login page after registering or receiving a new password. After successfully logging in, they can access the main application page [8].

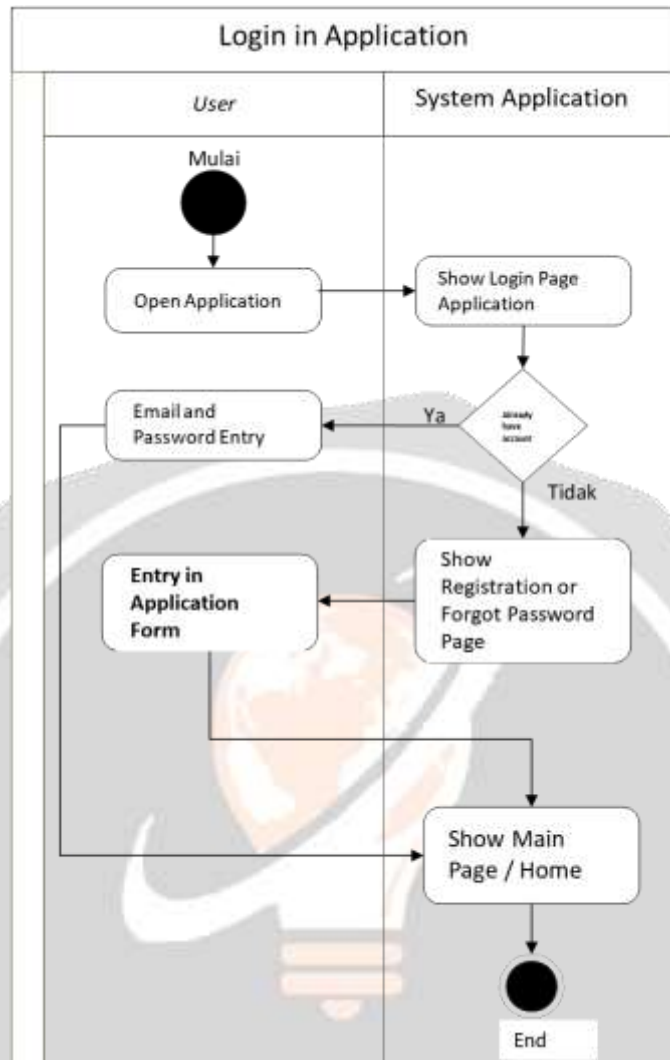


Fig -2 Activity Diagram User Accesses Profile Page on Application

Based on the ideas generated, a design that can be tested will be created at the prototype stage. This prototype helps developers validate user needs during the prototype stage before making the application. The Figma program is a supporting tool in the prototyping design process to create designs ranging from wireframes to Hi-Fi prototypes. The Figma application, which can be accessed directly through a web browser without initially installing it, will be utilized in the early stages of creating a digital prototype.

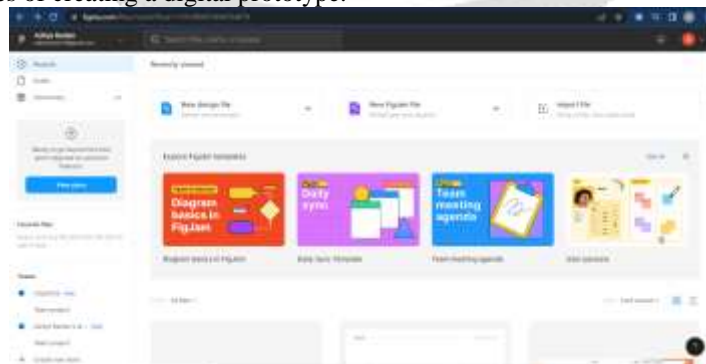


Fig -3 Figma Web Application Main Page

Wireframes are created at the beginning of the prototype design process. One type of low-fidelity design is a wireframe. Before adding colours and images, a wireframe serves as the fundamental structure and layout for presenting a page's general notion and showing the program's functionality. Wireframes are made utilizing only solid lines and forms, as the name implies [9].

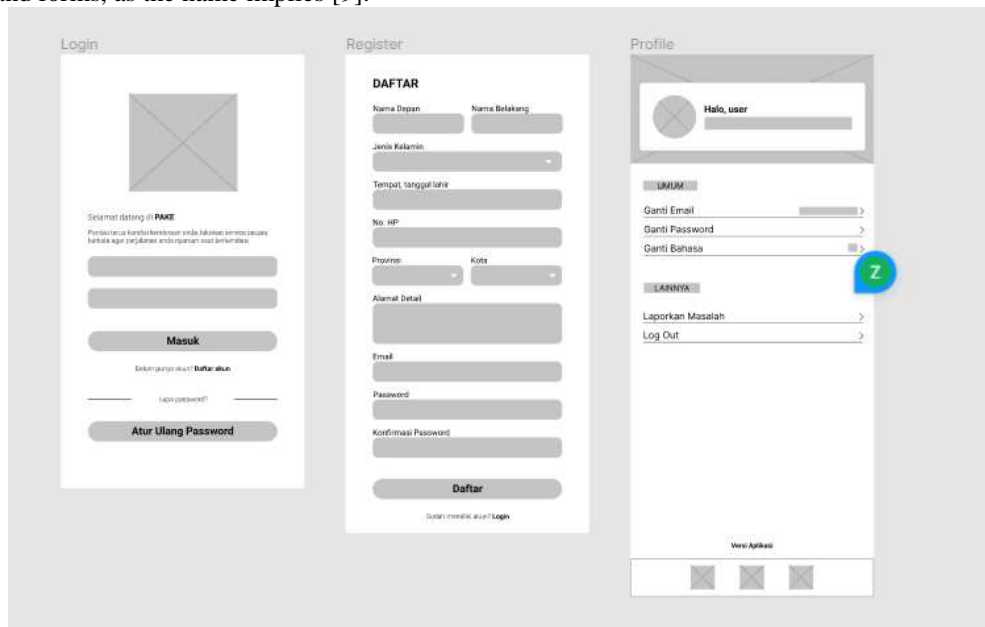


Fig -4 Figma Wireframe Design

Changing parts of the wireframes into Hi-Fi designs (High Fidelity) to include colours, graphics, and icons or logos is the next step in creating a prototype design. Create a sticker sheet on Figma as background information before making a Hi-Fi design. Hi-Fi methods require various elements, including typography, colour schemes, and other design elements. The vehicle maintenance application's prototype design is shown below.

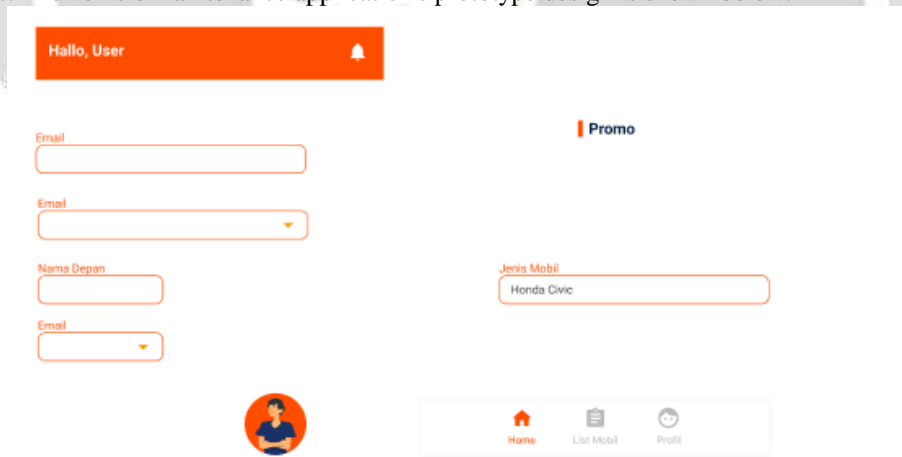
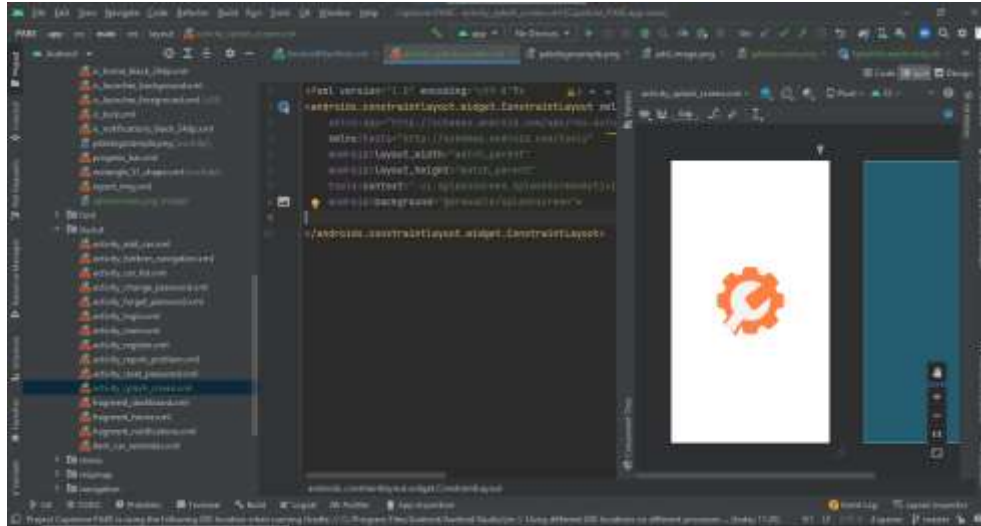


Fig -5 Figma Hi-Fi Design

#### 4. RESULTS AND DISCUSSION

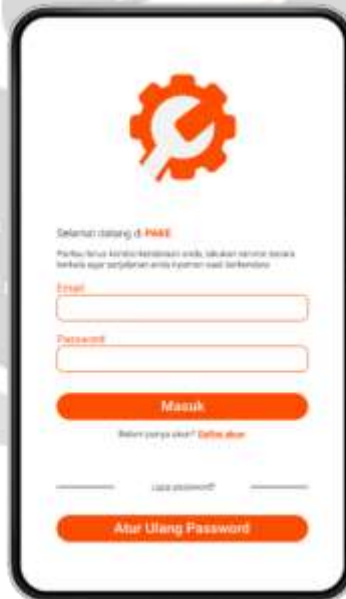
The next stage is using the Android Studio program to develop an Android application per the completed and fixed UX design. Kotlin is a programming language used in Android Studio to create applications. The first part of the application is SplashScreen. When a user first launches the application, the SplashScreen is an animated page that briefly displays the application's logo. The SplashScreen will serve as the application's identifying feature, just like a logo that moves into the program upon launch, a welcome screen that shows the program's icon, and a transition into the program itself. Prepare the logo and backdrop colour displayed on the SplashScreen initially while creating the SplashScreen.



**Fig -6** SplashScreen Design

The action bar can be hidden, the delay for switching activities can be set, transition animations can be set, and the activity displayed after the splash screen can also be set. These settings are made in the Kotlin programming portion of the application.

The vehicle maintenance application's login page appears as soon as the splash screen display has finished running when the application is opened. The application's logo can be seen on this page, along with a message that welcomes you and summarizes how to use vehicle maintenance. An email address and password are also required for the application's authentication process. Additionally, there are two buttons: "enter" to start the application and "Reset Password" to send the user to the Forgot Password page if necessary. Additionally, the phrase "Register account" has a hyperlink that can direct users to the account registration page. The login page is displayed in Figure 7 below.

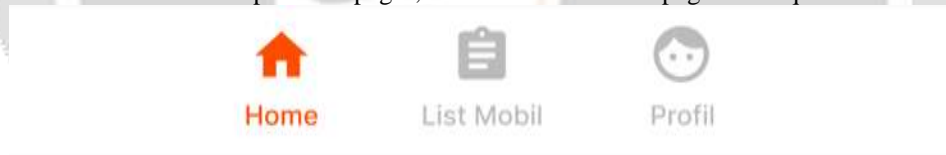


**Fig -7** Application Login Page

To implement the login page shown in Figure 7, activity and the login page's XML layout must first be created. Because of its flexibility, Constraint Layout is used in the login page layout to create sophisticated presentations. Custom Edit Text is used in the email and password sections so that writers can set character limits and passwords and disguise the visibility of passwords so that they are not visible to others as a form of security. For the registration page and forgetting the password, it is more or less the same as the process for creating the login page. Figure 8 below displays the registration page.

**Fig -8** Registration Page

The Navigation Bar, which works as a button to take the user to the pages available on the Navigation Bar, namely "Home," "vehicle list," and "profile," is at the bottom of the application page after each component of the feature is following the UX design, as shown in Figure 9. A fragment will be made, and afterwards, to display the navigation bar, which will also be used on multiple other pages, will be added to each page that requires it.



**Fig -9** Navigation Bar

The car list page shows a list of the cars that the user has added to the application. This page has a favourite car feature that lets you know which automobile is most interesting in the application. For a vehicle to be designated as a favourite, it must first be put on this page. Users can add their automobiles to the list of cars on this page. There is also an additional button that alters the intent of the car registration page. The application's car list page is shown in Figure 10 below.

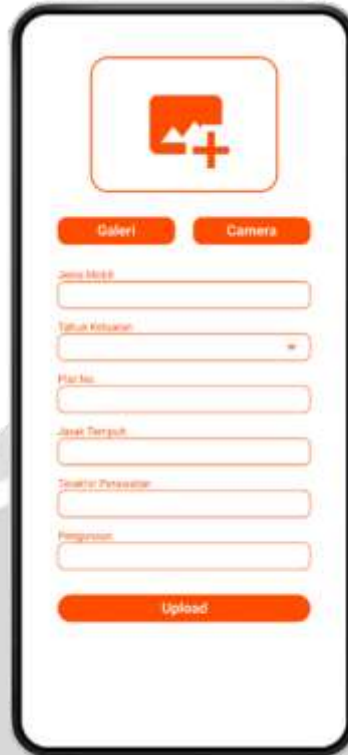




**Fig -10** Vehicle List Page

The car list page, which also generated activity and layout for the car list page in the form of XML, is shown in Figure 10. Also visible is the fact that the picture view is used to display photographs and buttons in the shape of an add button, a delete button in the form of a trash icon, and a button that says "Change Favorite Car." The program will change the intent of the add vehicle page by adding an explicit intent to the add button so that the user can add a car.

The Add Automobile page, a page with an express goal, helps add the user's car by entering the necessary information so that it can subsequently be presented on the car list. The add car page has the same features as the registration page, including an edit box for entering vehicle information, an image view, two camera buttons, and a gallery for uploading images of the user's vehicle. The intent camera is used on the add car page to make it simpler to control how the camera is utilized in the program. The add car page, also a part of the car list, is displayed in Figure 11 below.



**Fig -11** Add Car Page

The car list page shows a list of the cars that the user has added to the application. This page has a favourite car feature that can choose a favourite selection from the vehicle collection. For a vehicle to be designated as a favourite, it must first be put on this page. Additionally, an add button modifies the page's auto registration goal so visitors can add their vehicles to the list.

#### 4. CONCLUSIONS

1. In solving the problem, the Vehicle Maintenance application was developed as a vehicle maintenance application to reduce the number of road accidents.
2. In developing applications, the Kotlin programming language is used as the primary programming language compared to Java because the Kotlin programming language is more practical, safer from errors, and easier to understand.
3. In application development, there is a process of making application UX designs so that users can find out what is needed in an application.

#### 6. REFERENCES

- [1] R. G. Tiwari, A. Pratap Srivastava, G. Bhardwaj, and V. Kumar, "Exploiting UML Diagrams for Test Case Generation: A Review," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, London, United Kingdom, Apr. 2021, pp. 457–460. doi: 10.1109/ICIEM51511.2021.9445383.
- [2] I. M. Kisémbó *et al.*, "An Algorithm for Improving Email Security on the Android Operating System in the Industry 4.0 Era," *Journal of Engineering*, vol. 2021, pp. 1–8, Nov. 2021, doi: 10.1155/2021/4690611.
- [3] A. Nugraha and F. Aminur Rahman, "Android Application Development of Student Learning Skills in Era Society 5.0," *J. Phys.: Conf. Ser.*, vol. 1779, no. 1, p. 012014, Feb. 2021, doi: 10.1088/1742-6596/1779/1/012014.
- [4] M. N. M. Al-Faruq, S. Nur'aini, and M. H. Aufan, "PERANCANGAN UI/UX SEMARANG VIRTUAL TOURISM DENGAN FIGMA," *Walisongo J. Inf. Technol.*, vol. 4, no. 1, pp. 43–52, Aug. 2022, doi: 10.21580/wjit.2022.4.1.12079.
- [5] B. G. Mateus, M. Martinez, and C. Kolski, "Learning migration models for supporting incremental language migrations of software applications," *Information and Software Technology*, vol. 153, p. 107082, Jan. 2023, doi: 10.1016/j.infsof.2022.107082.

- [6] A. Gunawan and S. Y. Ln, “PELATIHAN ANDROID STUDIO UNTUK MENINGKATKAN KEMAMPUAN DOSEN DAN TENAGA PENDIDIK DALAM BIDANG TEKNOLOGI INFORMASI DI UNIVERSITAS BUNDA MULIA JAKARTA,” *JUR.PKM*, vol. 4, no. 5, p. 451, Nov. 2021, doi: 10.30998/jurnalpkm.v4i5.10837.
- [7] W. Xiao, H. He, W. Xu, X. Tan, J. Dong, and M. Zhou, “Recommending good first issues in GitHub OSS projects,” in *Proceedings of the 44th International Conference on Software Engineering*, Pittsburgh Pennsylvania, May 2022, pp. 1830–1842. doi: 10.1145/3510003.3510196.
- [8] R. Elmansouri, S. Meghzili, and A. Chaoui, “A UML 2.0 Activity Diagrams/CSP Integrated Approach for Modeling and Verification of Software Systems,” *csci*, vol. 22, no. 2, Apr. 2021, doi: 10.7494/csci.2021.22.2.3478.
- [9] B. D. Jaya and E. P. Agustini, “PERANCANGAN WIREFRAME USER INTERFACE SEMESTA BACA MENGGUNAKAN FIGMA,” p. 15.

